Sandia National Laboratories

Albuquerque, New Mexico 87185

date: February 24, 1993

to: Distribution

from: G. D. Sjaardema, 1562

subject: *EXOSYM*: A program for "enhancing" results visualization.

Introduction:

A program called *EXOSYM* has been written to help analysts produce more realistic looking visualizations of analysis results and models. *EXOSYM* reads as input a three-dimensional finite element mesh or results file in EXODUS [1] format and will mirror the geometry and results about the specified coordinate planes. An example may make this more clear.

Assume that you are analyzing a spherical object that, due to the symmetry of the load, requires that only 1 octant of the sphere be analyzed to fully capture the response of the object. However, to fully explain the problem that was analyzed, it is much easier to show the response of the entire sphere. Previously, your only two alternatives were to:

- mesh and analyze the entire object which increases the memory and analysis time by a factor of eight, or
- mesh and analyze one octant of the object and try to explain to your "audience" that the response you are showing actually models the entire object and they just have to "imagine" that the body appears in all of the octants.

Although the second alternative is not that bad and many of our customers understand symmetry planes, *EXOSYM* gives you a relatively painless method to increase the realism of your analyses.

Execution:

EXOSYM is installed in the SEACAS [2] system and is executed similar to any of the other translator programs:

`exosym -transform=`*octant_spec* **`-Material=`***mat_offset input_file output_file*

where, *input_file* is the input exodus file, *output_file* is the translated output exodus file, *octant_spec* is an 8 character string which specifies how the translation is to be performed, and *mat_offset* is the increment that will be added to each input material ID for each replicated output material ID. Each character position in *octant_spec* corresponds to an

octant in the three-dimensional coordinate space. The origin of the space is assumed to be at X=Y=Z=0.0. Table 1 describes the numbering of the octants. Each character position in

Table 1: Description of EXOSYM Octants

Octant	*X Coord*	*Y Coord*	*Z Coord*
1	positive	positive	positive
2	negative	positive	positive
3	negative	negative	positive
4	positive	negative	positive
5	positive	positive	negative
6	negative	positive	negative
7	negative	negative	negative
8	positive	negative	negative

the string can be either an ‘X’, a ‘1’, or a ‘0’. The ‘X’ signifies that the original body lies in that octant. A ‘1’ signifies that the body will be replicated into that particular octant and a ‘0’ signifies that the body will not be replicated into that particular octant. Only one ‘X’ may appear in the *octant_spec* string. A few examples will hopefully clarify this:

- -transform=X0101010 -- Original body in octant 1, replicate to octants 3, 5, and 7.
- -transform=X1000000 -- Original body in octant 1, replicate to octant 2. This is simply a mirroring about the Y-Z coordinate plane.
- -transform=0001000X -- Original body in octant 8, replicate to octant 4. This is simply a mirroring about the X-Y coordinate plane.

Note that the octant notation is simply a means of specifying mirroring about any combination of the coordinate planes. Your original object does not have to lie entirely in a single octant to use *EXOSYM*. In fact, for man cases the original body may lie in 4 octants and you simply want it replicated into the other 4 octants.

Table 1 can also be used to determine the transformations that will be applied to the original body to replicate it into the selected octant(s) by simply multiplying the original row by the selected row. For example, for the case -transform=0001000X, multiply the values in row 8 (+1, -1, -1) by the values in row 4 (+1, -1, +1) to get (+1, +1, -1). This means that the transformation will be performed by negating all Z-coordinate values and simply copying the other coordinate values.

EXOSYM essentially replicates the input geometry the requested number of times without joining the created bodies to the original body. Therefore, each generated element block must have a unique material ID. *EXOSYM* increments each input material ID by the *mat_offset* value specified by the **`-Material`** option. This number may take some

experimentation to determine since, for example, for a 8 material input file that is replicated into 8 octants, each of the 64 final element blocks must have a different material ID.

Nodal Results Transformation:

EXOSYM assumes that the nodal results variables are in groups of three, each member of the group corresponding to the X, Y, and Z component of the variable vector. The first group of three is assumed to be the nodal displacements. If the number of nodal variables is not a multiple of three, *EXOSYM* will issue a fatal error message and stop. Variables are transformed in the same way that the nodal coordinates are transformed. For example, if the transformation is such that the X- and Z-coordinates are negated, then the first and third nodal variable in each variable vector group will also be negated. Simply put, if the X-coordinate is negated, then DISPLX, VELX, and ACCLX will also be negated.

Element Results Transformation:

This version of EXOSYM simply copies all of the input element variables to the output element variables. This is clearly not the correct thing to do for tensor variables such as stress and strain, but it works for scalar variables such as von Mises stress, temperature, and equivalent plastic strain.

This limitation is planned to be removed in a future version of *EXOSYM*.

Limitations:

- Only three-dimensional exodus files are supported[1].
- Nodal results are assumed to be vectors with three components corresponding to the three coordinate directions.
- Tensor Element quantities are **not** translated correctly.
- Element blocks are created with new material IDs rather than joining the created blocks to the original blocks.
- All transformations occur about coordinate planes centered on the X, Y, or Z axis. There is no way to specify both an offset and a mirroring.[2]
- This is a new code that is likely to have some bugs. Proceed with caution.
- *EXOSYM* appends its program name and QA string to the QA records in the output EXODUS file.[3]

Example:

The following figure shows the result of a simple execution of *EXOSYM*. The input mesh is simply a 27-element cube in octant 5. The command line used to create the output mesh is: **`exosym -transform=0101X010 input.g output.g`**

---

1. The BLOT view command can be used to do a similar function for two-dimensional exodus files.
2. This capability will be added to the next version if there is sufficient need. I have the method for implementing this designed.
3. This is a limitation only if you are trying to impress someone with the size of the calculation you performed since they will be able to see that EXOSYM did something to the mesh.

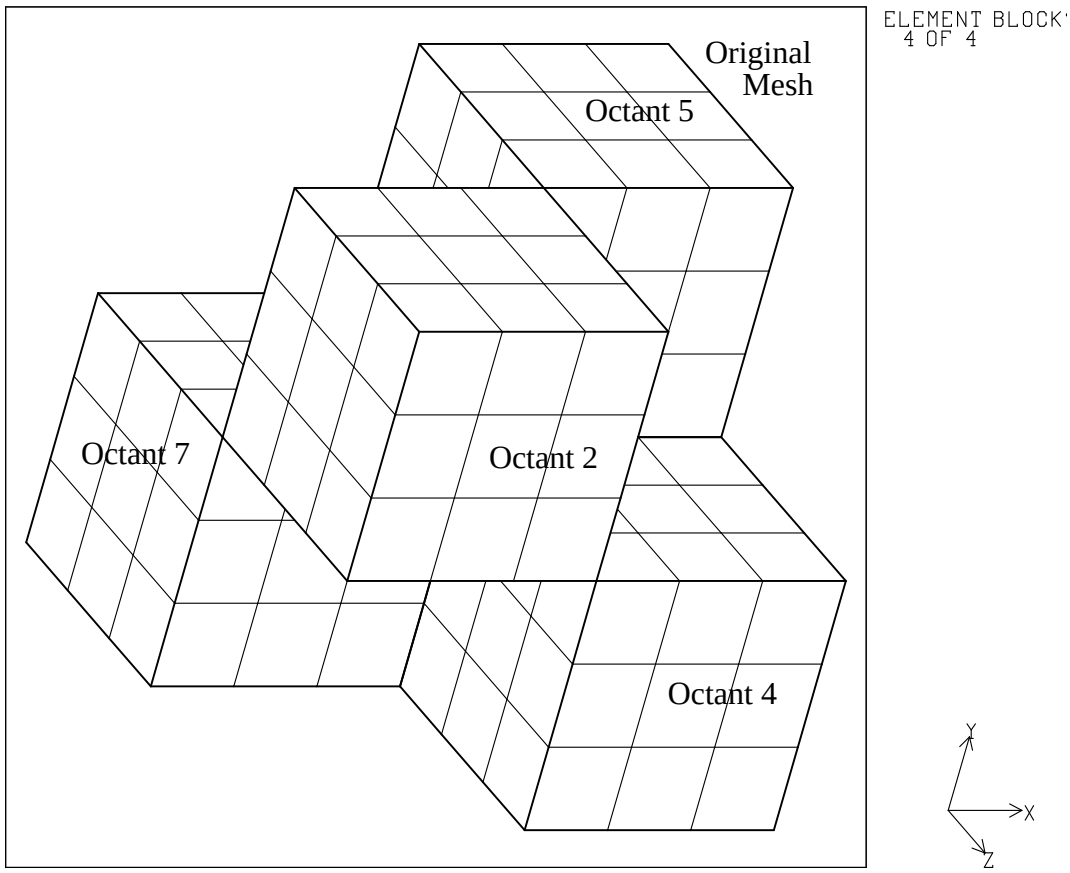

Figure 1. Example EXOSYM Output

References

[1] W. C. Mills-Curran, A. P. Gilkey, and D. P. Flanagan, “EXODUS: A Finite Element File Format for Pre- and Post-processing,” SAND87–2977, Sandia National Laboratories, Albuquerque, NM, September 1988.

[2] G. D. Sjaardema, “Overview of the Sandia National Laboratories Engineering Analysis Code Access System,” SAND92–2292, Sandia National Laboratories, Albuquerque, NM, January 1993.

Distribution:

13	1425	J. H. Biffle & staff
1	1431	J. M. McGlaun
1	1431	K. G. Budge
1	1431	J. S. Peery
1	1432	W. T. Brown
15	1434	D. R. Martinez & staff
1	1500	D. J. McCloskey
1	1501	C. W. Peterson
1	1502	P. J. Hommert
1	1503	L. W. Davison
1	1511	J. S. Rottler
1	1511	D. K. Gartling
1	1511	M. W. Glass
1	1511	P. L. Hopkins
1	1511	M. J. Martinez
1	1511	P. A. Sackinger
1	1511	P. R. Schunk
1	1512	A. C. Ratzel
1	1513	R. D. Skocypec
1	1513	R. G. Baca
1	1513	B. L. Bainbridge
1	1513	R. E. Hogan, Jr.
1	1513	J. L. Moya
1	1551	W. P. Wolfe
1	1552	C. E. Hailey
1	1553	W. L. Hermina
1	1554	W. H. Rutledge
15	1561	H. S. Morgan & staff
13	1562	R. K. Thomas & staff
1	1562	G. D. Sjaardema
1	2565	S. T. Montgomery
1	6313	J. Jung
1	6411	A. S. Benjamin
1	6522	J. D. Miller
6	8741	G. A. Benedetti & staff
1	8742	M. R. Birnbaum
1	8742	J. J. Dike
1	8742	L. I. Weingarten
5	8743	M. L. Callabresi & staff